

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 2 of 12

Attorney's Docket No.: 16366-012001

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A message queue management system configured to manage a plurality of queues including a first message queue of a first data store type and a second message queue of a second data store type, wherein the first message queue and the second message queue exhibit consistent behavior across a plurality of computer mediums, the message queue management system comprising:

a journalling module configured to track and manage message states of messages stored in the first message queue and the second message queue including a journalled state whereby journalled messages are retained in the first message queue and the second message queue;

a once-only message processing module configured to track a plurality of message states to prevent re-processing of previously processed messages stored in the first message queue and the second message queue;

an interoperability module configured to coordinate interactions among the plurality of queues enabling a distributed workflow;

a recovery module configured to perform data recovery on the plurality of queues upon the event of a failure;

a central managing module configured to provide single point control of the plurality of queues, wherein the single point control includes controlling a plurality of publishers and a plurality of subscribers; and

a load balancing module configured to designate a subset of the plurality of subscribers to at least the first message queue in a subscriber pool such that messages in the first message queue are processed by a first available subscriber in the subscriber pool and to enable the

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 3 of 12

Attorney's Docket No.: 16366-012001

subscribers to subscribe to at least the first message queue in parallel with subscribers not in the subscriber pool.

2. (Cancelled)

3. (Currently Amended) A message queuing system for computer networks that provides reliable communication, ensures non-repetitive delivery, and is implemented in a medium-independent environment, the message queuing system comprising:

a message queue configured to run on one of a plurality of environments and to store messages received from message publishers and monitor retrieval by a plurality of message subscribers wherein a message is retained in the message queue after at least one of the plurality of message subscribers has accessed the message to permit additional access to the message and to assist in at least one of archiving the message, manual distributing the message, editing the message, or resending the message;

a message state module configured to track message states of a message wherein message state information indicates whether the message has been made available to and retrieved by at least one of the plurality of the message subscribers;

a recording module configured to selectively retain the message state information related to corresponding messages at least until the message is removed from the message queue; and

an interface module configured to receive and send a plurality of types of data objects in messages and to communicate with a plurality of types of data messaging systems that act as message publishers or message subscribers.

4. (Original) The message queuing system of Claim 3, wherein at least one of the plurality of the data messaging systems acts as both a message publisher and a message subscriber.

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 4 of 12

Attorney's Docket No.: 16366-012001

5. (Original) The message queuing system of Claim 3, wherein the message queue is further configured to provide load balancing.

6. (Original) The message queuing system of Claim 5, wherein the load balancing is implemented using subscriber pooling.

7. (Original) The message queuing system of Claim 3, wherein the message state information is used to ensure that messages are acquired in the designated order by at least one of the plurality of message subscribers.

8. (Currently Amended) The message queuing system of Claim 3, wherein ~~a~~the message is intended for a single message subscriber.

9. (Currently Amended) The message queuing system of Claim 3, wherein ~~a~~the message is intended for a plurality of message subscribers.

10. (Original) The message queuing system of Claim 3, wherein the message is deleted from the message queue during a designated clean up process.

11. (Original) The message queuing system of Claim 3, wherein the message is retained in the message queue to assist in recovery of messages during a system failure.

12-13. (Cancelled)

14. (Original) The message queuing system of Claim 3, wherein the message state information includes message publisher state information and corresponding message subscriber state information.

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 5 of 12

Attorney's Docket No.: 16366-012001

15. (Original) The message queuing system of Claim 14, wherein the message state information is stored within the message.

16. (Original) The message queuing system of Claim 3, wherein the message states reflect the message publisher's status of the message in relation to the intended message subscribers and an individual message subscriber's status of the message in relation to the corresponding individual message subscriber.

17. (Original) The message queuing system of Claim 3, wherein the message state includes a sequence number, publisher state information, and subscriber state information.

18. (Original) The message queuing system of Claim 17, wherein the publisher state information includes information relating to an "unrevealed" state, information relating to a "revealed" state, information relating to an "unavailable" state, and information relating to a "journalled" state.

19. (Original) The message queuing system of Claim 17, wherein the subscriber state information includes information relating to a "get" state, information relating to a "fetched" state, information relating to an "unusable" state, information relating to an "expired" state, information relating to an "inprocess" state, information relating to a "done" state, and information relating to a "journalled" state.

20. (Original) The message queuing system of Claim 3, wherein the message queue is configured to interface with a plurality of different database protocols.

21. (Original) The message queuing system of Claim 3, wherein at least one of the types of a plurality of data messaging systems is a database.

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 6 of 12

Attorney's Docket No.: 16366-012001

22. (Original) The message queuing system of Claim 3, wherein at least one of the types of a plurality of data messaging systems is a flat file directory.

23. (Original) The message queuing system of Claim 3, wherein at least one of the types of a plurality of data messaging systems is a second message queuing system.

24. (Original) The message queuing system of Claim 3, wherein at least one of a plurality of data objects is a database record.

25. (Original) The message queuing system of Claim 3, wherein at least one of a plurality of data objects is an electronic mail message.

26. (Currently Amended) The message queuing system of Claim 3, wherein at least one of a plurality of data objects is a web page.

27. (Currently Amended) A method of message queuing among a plurality of message publishers and a plurality of message subscribers, the method comprising:

receiving a data object from a message publisher intended for at least one message subscriber;

processing the data object to create a corresponding message wherein the corresponding message includes the data object and message information, where the message information includes a message identifier and a message header and the message header includes message state information for the message publisher and the at least one message subscriber;

storing the message in a message queue;

tracking the status of the message from ~~the~~ a perspective of the message publisher;

tracking the status of the message from ~~the~~ a perspective of the at least one message subscriber; and

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 7 of 12

Attorney's Docket No.: 16366-012001

selectively retaining the message ~~to remain~~ in the message queue after the message has been retrieved by the at least one message subscriber.

28. (Original) The method of Claim 27, wherein the message queue is configurable to run on a plurality of platforms.

29. (Original) The method of Claim 27, wherein the message queue is configurable to provide load balancing.

30. (Original) The method of Claim 27, wherein the message queue is configurable to provide subscriber pooling.

31. (Original) The method of Claim 27, wherein the message information includes a message identifier and a message header.

32-33. (Cancelled)

34. (Currently Amended) The method of Claim ~~33~~27, wherein the message information for the message publisher includes information related to message publisher states wherein the message publisher states include a "put" state, an "unavailable" state, a "revealed" state, and a "journalled" state.

35. (Currently Amended) The method of Claim ~~33~~27, wherein the message information for the at least one message subscriber includes information related to message subscriber states wherein the message subscriber states include a "get" state, a "fetched" state, an "unusable" state, an "expired" state, an "inprocess" state, a "done" state, and a "journalled" state.

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 8 of 12

Attorney's Docket No.: 16366-012001

36. (Currently Amended) The method of Claim 3227, wherein the message state information is stored in a log that includes time stamps for ~~the~~ corresponding message states.

37-39. (Cancelled)

40. (Currently Amended) A system for message queuing among a plurality of message publishers and a plurality of message subscribers, the system comprising:

a message queue module configured to communicate with a plurality of message publishers and a plurality of message subscribers and to receive a data object from a message publisher intended for at least one of a plurality of message subscribers;

a data object processing module configured to process the data object to create a corresponding message wherein the corresponding message includes the data object and message information, the message information including a message identifier and a message header, the message header including message state information for the message publisher and the at least one of the plurality of message subscribers;

a publisher tracking module configured to track ~~the~~ a status of the message from ~~the~~ a perspective of the message publisher;

a plurality of subscriber tracking modules configured to track ~~the~~ a status of the message from ~~the~~ a perspective of the at least one of a plurality of intended message subscribers; and

a message tracking module configured to selectively retain the message ~~to remain~~ in the message queue after the message has been retrieved by the at least one of a plurality of intended message subscribers.

41. (Original) The system of Claim 40, wherein the message queue module is further configured to provide load balancing.

42. (Original) The system of Claim 40, wherein the message queue module is further configured to provide subscriber pooling.

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 9 of 12

Attorney's Docket No.: 16366-012001

43-44. (Cancelled)

45. (Original) The system of Claim 40, wherein the message information includes message state information.

46. (Original) The system of Claim 45, wherein the message state information is stored in a log that includes time stamps for the corresponding message states.

47. (Currently Amended) A method for queuing messages received from a publisher for retrieval by a subscriber comprising:

storing messages using a queue and a queue interface module wherein the queue interface module is configurable to interface with a plurality of database protocols;

extending message storage beyond the a time in which the a message has been retrieved by all intended subscribers; and

tracking message states, including tracking information related to message publisher states and message subscriber states, to enforce once-only delivery;

wherein the message publisher states include a "put" state, an "unavailable" state, a "revealed" state and a "journalled" state.

48-49. (Cancelled)

50. (Currently Amended) The method of Claim ~~48~~⁴⁷, wherein the message ~~publisher~~ subscriber states include a "get" state, a "fetched" state, an "unusable" state, an "expired" state, an "inprocess" state, a "done" state, and a "journalled" state.

51. (Cancelled)

Applicant : Beckman et al.
Serial No. : 09/704,016
Filed : November 1, 2000
Page : 10 of 12

Attorney's Docket No.: 16366-012001

52. (Original) A method for restoring message communication between a first message queue implemented using a first type of data store and a second message queue implemented using a second type of data store, the method comprising:

- processing the first message queue to locate messages that have not been revealed;

- for messages that have not been revealed:

- extracting path information to find message parents of the message;

- inspecting the message parents' states in the second message queue;

- marking message parents as "done" and the message "revealed" if at least one of the parent messages is marked "done;"

- marking message parents "clean" and the message "deleted" when the parent messages are not marked "done;" and

- marking the message in the second message queue "revealed" if there are no parents;

- recovering messages marked "clean;"

- processing the second message queue to find messages that are marked "fetched" but not "done"; and

- marking the messages that are marked "fetched" but not "done" as "clean."